```
#!/usr/bin/env python
#                                       -*- Mode: Python -*-

R"""palmhali converts a pilot-addresses file to a mh aliasses file
```

Usage: palmhali [-p port | -f pa-file]

palmhali converts a (palm) pilot-addresses file (pa-file) to a mh aliasses file. Default the pilot addresses file will be fetched directly from your palm device (overide this with the -f option). You have to install the pilot-addresses program (contained in the pilot-link package [1]) to use palmhali. The -p option can be used t specify the device file (port) to communicate with the palm device.

[1] ftp://ryeham.ee.ryerson.ca/pub/PalmOS/

(C) Copyright Anders Andersen 2000 (see COPYING for details).

```
"""

# Inserted by rcs (cvs)
rcs_author = "$Author: aa $"
rcs_date = "$Date: 2000/03/01 21:09:37 $"
rcs_revision = "$Revision: 1.6 $"

# Load some module
import sys, os, getopt, re, string

# Pilot-addresses command (first %s is the optional -p option and
# second %s is the name of the tmp file)
pa = "pilot-addresses %s -w %s"

# Temporal file (to store the addresses from the pilot (not used with
# the -f option)
if "TMP" in os.environ.keys():
    tmp = "%s/.adr" % (os.environ['TMP'],)
elif "TEMP" in os.environ.keys():
    tmp = "%s/.adr" % (os.environ['TEMP'],)
else:
    tmp = "/tmp/.adr"

# Fetch arguments
try:
    opts, args = getopt.getopt(sys.argv[1:], "p:f:")
except getopt.error, msg:
    sys.stderr.write("\npalmhali: %s.\n\n" % (msg,))
    sys.stderr.write(__doc__)
    sys.exit(1)


# No extra arguments
if len(args) > 0:
    sys.stderr.write("\npalmhali: no extra arguments please.\n\n")
    sys.stderr.write(__doc__)
    sys.exit(1)

# Parse arguments
opt = addrfile = ""
for o, v in opts:
    if o == "-p":
        opt = "-p %s" % (v,)
    if o == "-f":
        addrfile = v
```

```
# Select (and maybe create) the the pa-file
if not addrfile:
    if os.system(pa % (opt, tmp)):
        sys.stderr.write('\npalmhali: unable to "%s".\n\n' % (pa % (opt, tmp),))
        sys.stderr.write(__doc__)
        sys.exit(1)
else:
    tmp = addrfile

charmap = {
    "ð" : "d",
    "é" : "e",
    "É" : "E",
    "æ" : "a",
    "Æ" : "A",
    "ø" : "o",
    "Ø" : "O",
    "ö" : "o",
    "Ö" : "O",
    "å" : "a",
    "Å" : "A"}

def mapchars(str):
    for c in charmap.keys():
        if string.find(str, c) != -1:
            str = string.replace(str, c, charmap[c])
    return str

def splitstrlist(str):
    if str == "nil":
        return []
    else:
        return string.split(str[2:-2], '" "')

def rmws(str):
    return re.sub("\s*", "", str)

lineexp = re.compile(
    r'"(.*?)","(.*?)","(.*?)","(.*?)","(.*?)","(.*?)","(.*?)","(.*?)",' +
    r'"(.*?)","(.*?)","(.*?)","(.*?)","(.*?)","(.*?)","(.*?)","(.*?)",' +
    r'"(.*?)","(.*?)","(.*?)","(.*?)"', re.MULTILINE | re.DOTALL)

ln = 1
fn = 2
com = 4
email = 9
ali = 18        # I use custom 4 for mail aliases

# Read the pa-file
used = []
eaddrs = []
pos = 0
addr = open(tmp).read()
if not addrfile:
    os.remove(tmp)
end = len(addr)
while end > pos:
```

```
# Group a record (line)
m = lineexp.match(addr, pos)
if m:

    # Go to next record
    pos = m.end() + 1

    # Create email list
    elist = string.split(rmws(m.group(email)), ",")

    # Skip records without email addresses
    if elist[0] == "":
        continue

    # Insert aliases
    for e in elist:

        # Pre-process strings
        alin = string.lower(rmws(m.group(ali)))
        fnn = string.split(string.lower(mapchars(m.group(fn))))
        lnn = string.lower(mapchars(m.group(ln)))

        # Create name
        if m.group(fn):
            name = mapchars(m.group(fn))
            if m.group(ln):
                name = name + " " + mapchars(m.group(ln))
        else:
            name = mapchars(m.group(com))

        # Email alias at custom 4?
        if alin:
            aka = alin

        # Create alias from the name
        elif fnn:
            aka = fnn[0]
            if lnn:
                aka = aka + lnn[0]

        # Create alias from company name
        else:
            aka = re.sub("\W+", "-",
                string.lower(mapchars(m.group(com))))
            if not aka:
                sys.stderr.write("Ignored %s\n" % (e,))
                break
            else:
                while aka[-1] == "-":
                    aka = aka[:-1]

        # The alias has to be unique
        if aka in used:
            if e in eaddrs:
                break
            c = ord('A')
            aka = aka + chr(c)
```

```
            while aka in used:
                c = c + 1
                aka = aka[:-1] + chr(c)

        # Append alias
        used.append(aka)
        eaddrs.append(e)
        if len(aka) < 7:
            space = "\t\t"
        else:
            space = "\t"
        if name:
            sys.stdout.write("%s:%s%s <%s>\n" % (aka, space, name, e))
        else:
            sys.stdout.write("%s:%s%s\n" % (aka, space, e))

# None record (error or end)
else:
    break
```